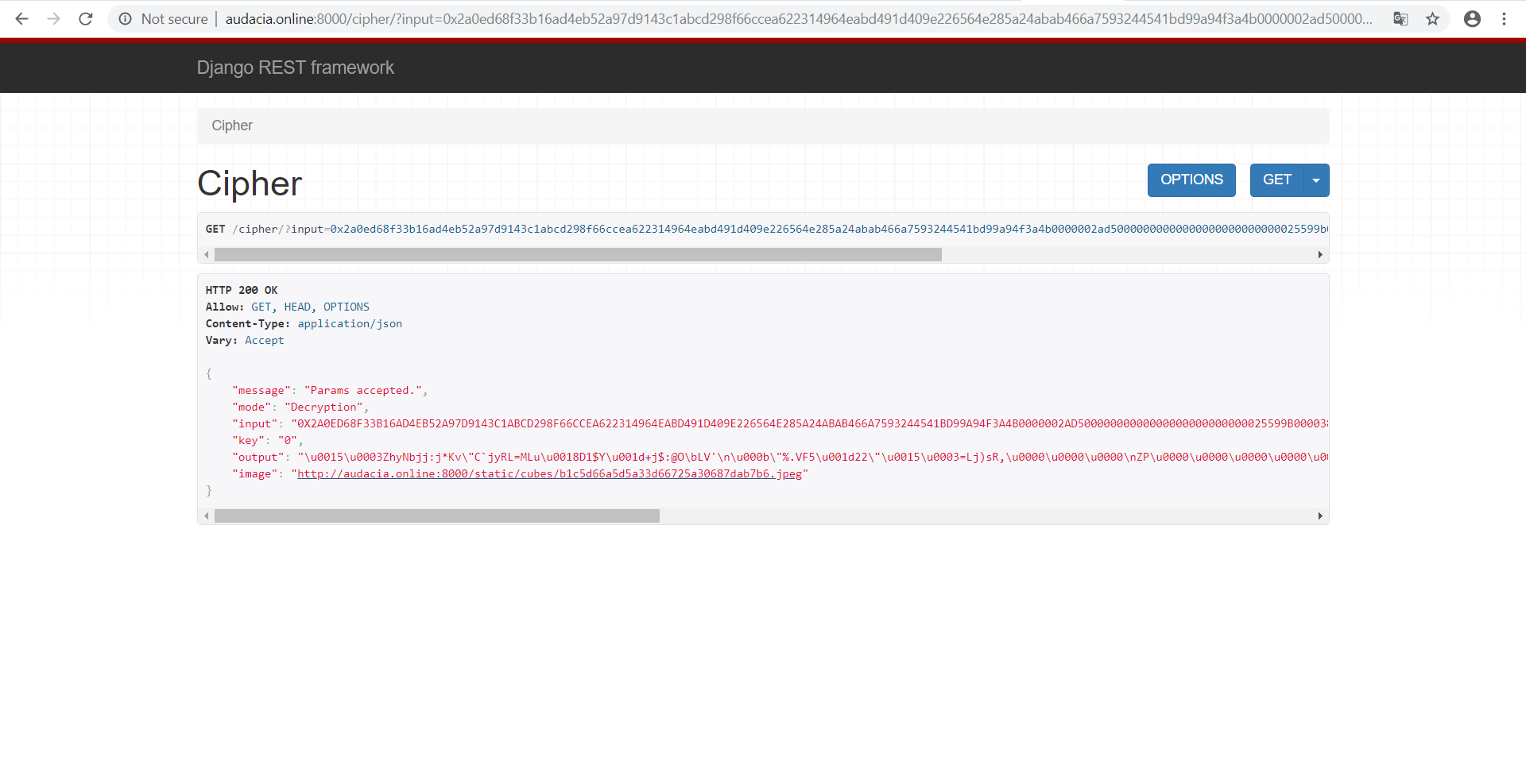
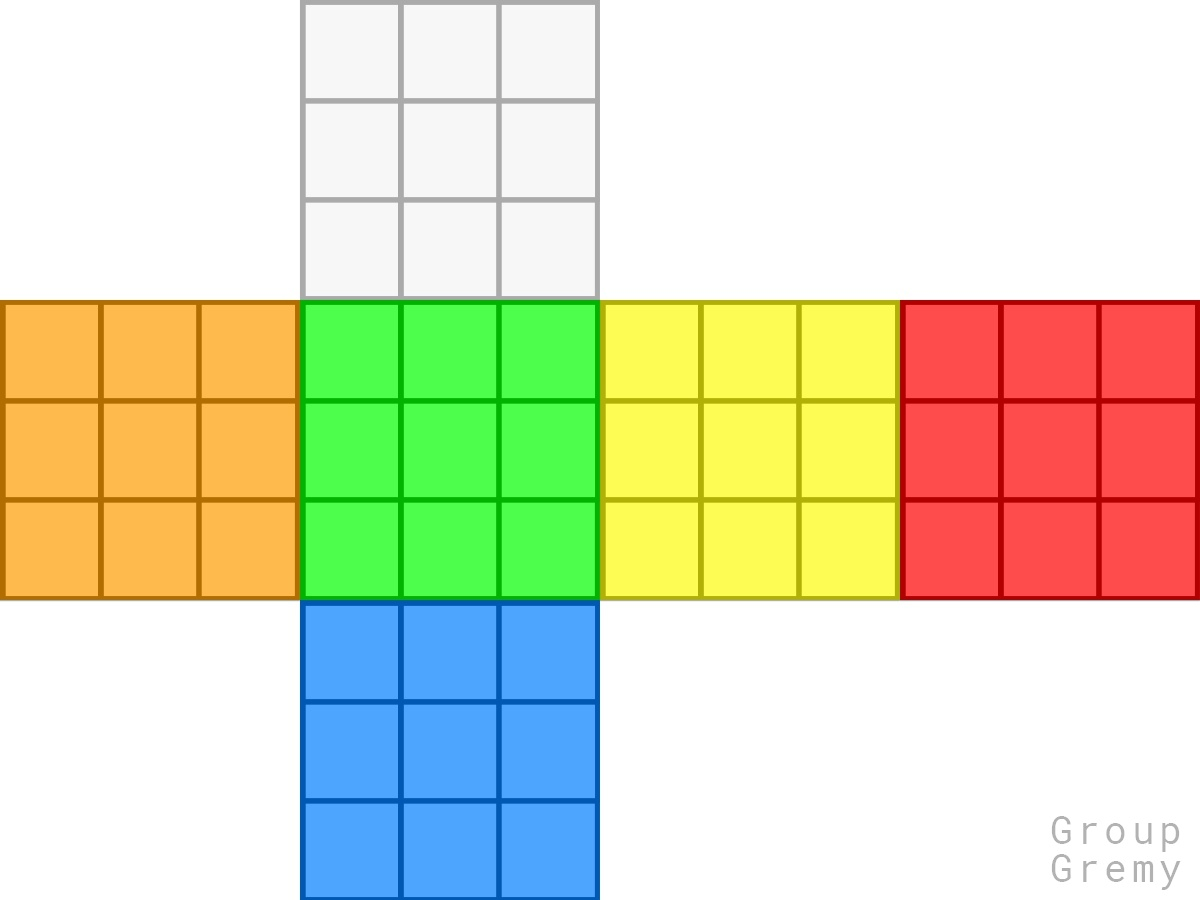
**Gremy Challenge Solution  
Gabriel Chan Zheng Yong (1002820)**

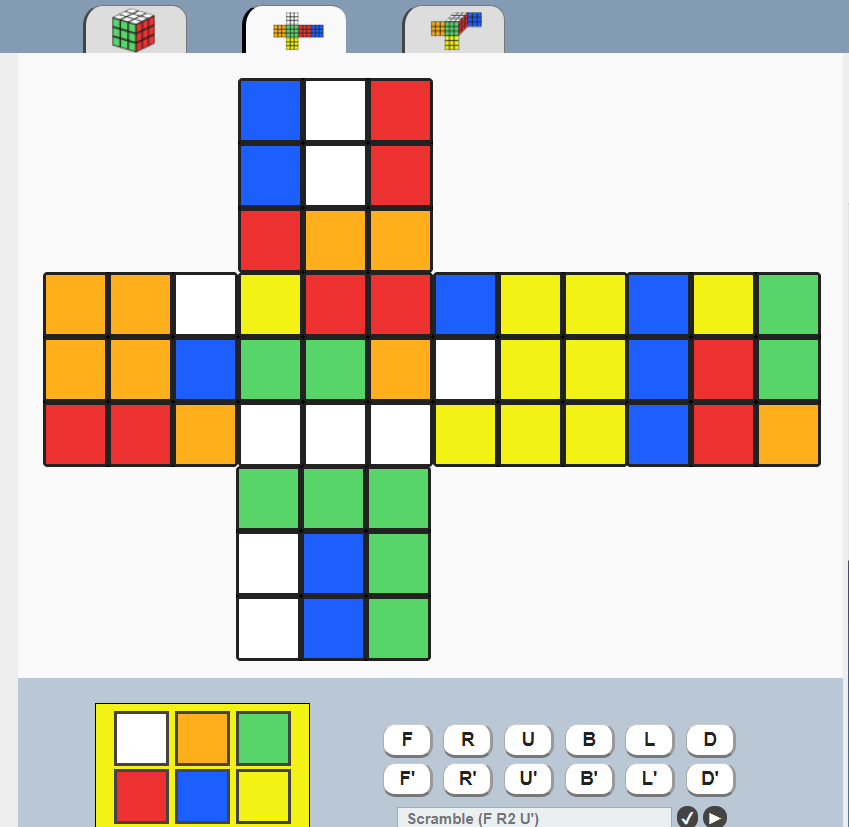
I was given a brief, a file with the enigma code template, the final state of the Rubik's Cube and a hex number to start the challenge.

I spent some time figuring out the different items that were given to me, including the test server. I figured that I had to start with the Rubik’s Cube problem. When I submitted the hex number to the server, I got a clean Rubik’s Cube. So I tried different capital letter keys to turn the cube and figure out how to get to the state given to me in the brief. (<http://audacia.online:8000/cipher/?input=0x2a0ed68f33b16ad4eb52a97d9143c1abcd298f66ccea622314964eabd491d409e226564e285a24abab466a7593244541bd99a94f3a4b0000002ad50000000000000000000000000025599b000038890052000000000000000000000000000000000000000035000062000000&key=0&mode=D&layer=2>)





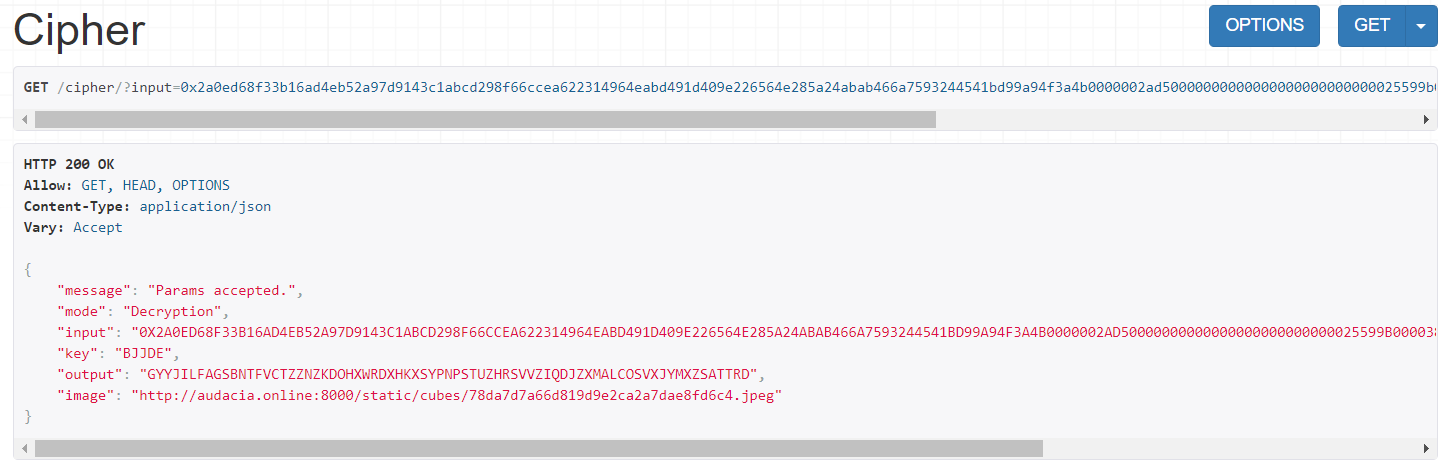
Next, I used a [Rubik’s Cube solver](https://rubiks-cube-solver.com/) to get the steps from a solved Rubik’s Cube to the state shown in the picture given to me in the brief.

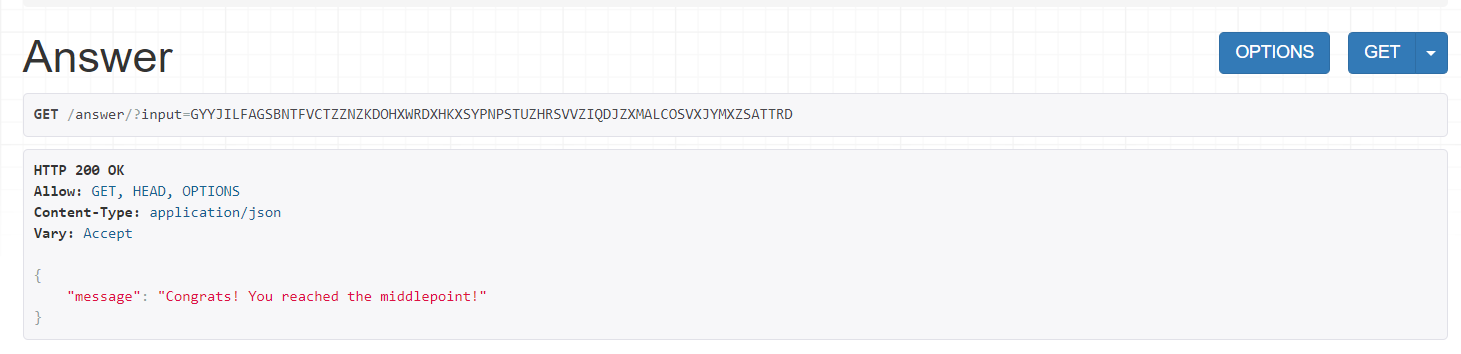


I matched the solution I got from the solver to the steps that is needed on their server. For example, if I needed to turn a certain face, I tried different letters in team Gremy’s server that would allow me to do that. I got this list of turns: **BJJDE**. This gave me a new ciphertext: **GYYJILFAGSBNTFVCTZZNZKDOHXWRDXHKXSYPNPSTUZHRSVVZIQDJZXMALCOSVXJYMXZSATTRD**. I confirmed the ciphertext with the answer checker and it was correct.

(Turning the cube: <http://audacia.online:8000/cipher/?input=0x2a0ed68f33b16ad4eb52a97d9143c1abcd298f66ccea622314964eabd491d409e226564e285a24abab466a7593244541bd99a94f3a4b0000002ad50000000000000000000000000025599b000038890052000000000000000000000000000000000000000035000062000000&key=BJJDE&mode=D&layer=2>)

(Midpoint answer: <http://audacia.online:8000/answer/?input=GYYJILFAGSBNTFVCTZZNZKDOHXWRDXHKXSYPNPSTUZHRSVVZIQDJZXMALCOSVXJYMXZSATTRD>)

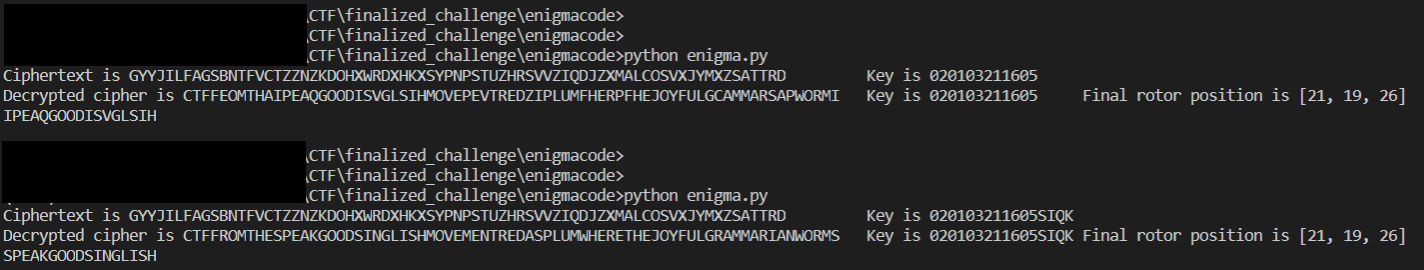




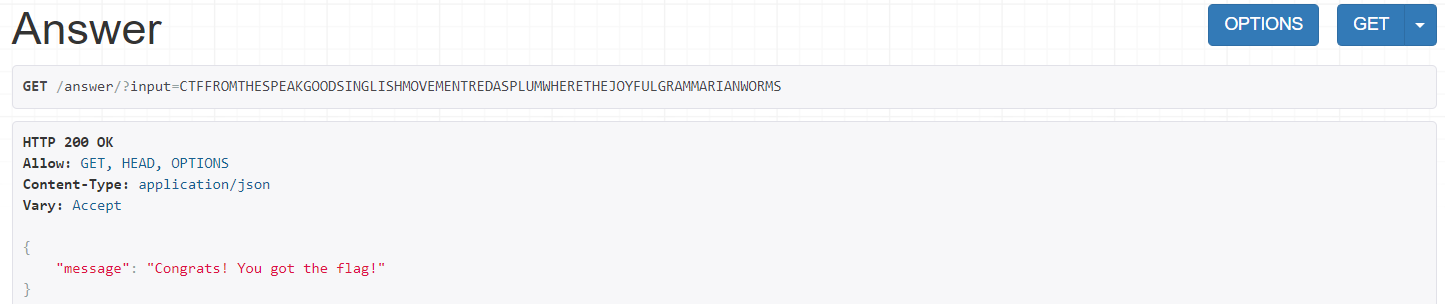
Next, I looked at the enigma python code template given to me. The hint given was to do a brute force for the first 4 letters “CTFF”. The enigma code given to me used a Rotor and Rotors object combined with a plugboard function to simulate the enigma machine. The hint further elaborated to do a partial brute force with just the rotors for ‘CTFF’. I took time to understand the code and then wrote python code to try out different combinations of keys without the plugboard. The key ”020103211605“ gave me the string “CTFFEOMTHAIPEAQGOODISVGLSIHMOVEPEVTREDZIPLUMFHERPFHEJOYFULGCAMMARSAPWORMI”.



Knowing that flag[10:27] gave me “SPEAKGOODSINGLISHMOVEMENT”, I switched 2 sets of letters (‘S’ & ‘I’, ‘Q’ & ‘K’) to get the key: “020103211605SIQK”. This gave me the flag “**CTFFROMTHESPEAKGOODSINGLISHMOVEMENTREDASPLUMWHERETHEJOYFULGRAMMARIANWORMS**”.



I checked with the online checker on their server.



The challenge was fun to try out. Most of my time was actually spent on trying to understand how to attempt the challenge rather than doing the challenge itself (things like testing the REST server given, understanding the enigma code). The hints given helped to solve the challenge much more easily, so if I did not have the hints I would have to spend more time trying to figure out how to break an enigma machine (they did give that their challenge started with ‘CTFF’ in their brief so it would be doable). The challenge helped me to understand that brute forcing only works if you know what the endpoint of your brute force is.